



JUSUF

Access to HPC at JSC

2022/05/17 | ANKE VISSER



Mitglied der Helmholtz-Gemeinschaft

JUSUF

jusuf.fz-juelich.de

JUSUF



- Jülich Support for Fenix
 - Component in the federated pan-European research e-infrastructure built up by BSC, CEA, CINECA, CSC, CSCS & JSC
 - Provisioned and operated as part of ICEI within the Human Brain Project, co-financed by EC
- Prime contractor: Atos
- Hybrid HPC/Cloud system particularly for interactive workloads
 - Compute partition for regular HPC jobs
 - Cloud partition with OpenStack for virtual machines and block storage
 - expected workloads: HBP platform & web services, databases/repositories, analytics
- Part of HPC infrastructure at JSC
 - JUWELS & JURECA-DC have similar hardware & software environments

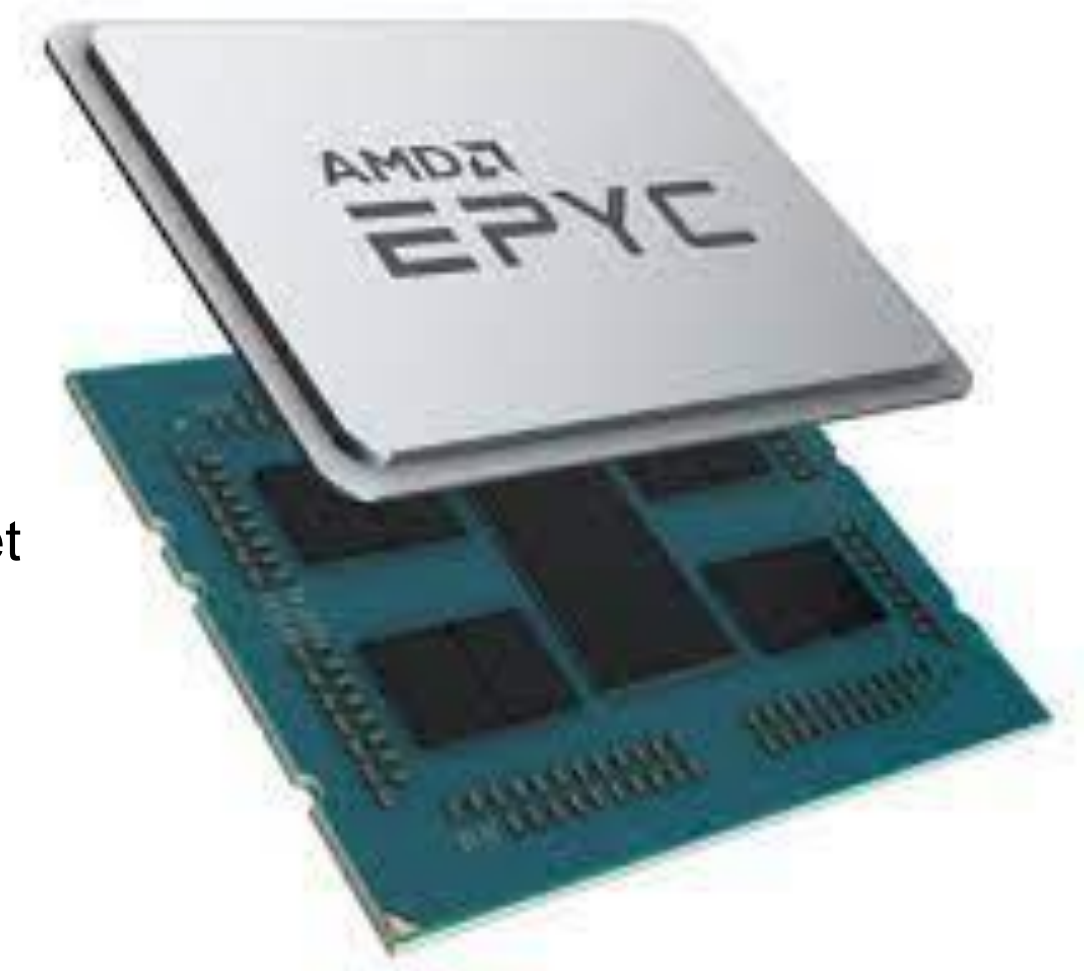
JUSUF SYSTEM OVERVIEW

- **144 *standard* compute nodes**
 - dual AMD EPYC 7742 processors, each with 64 cores, 2.25 GHz, 256 GB DDR4 RAM
- **61 *accelerated* compute nodes**
 - dual AMD EPYC 7742 processors, each with 64 cores, 2.25 GHz, 256 GB DDR4 RAM
 - *single Nvidia V100 'Volta' GPU with 16GB HBM2e*
- **4 *login* nodes (and 7 additional service nodes)**
 - dual AMD EPYC 7742 processors, each with 64 cores, 2.25 GHz, 256 GB DDR4 RAM
- Mellanox InfiniBand HDR full fat-tree interconnect network
- IBM Spectrum Scale (GPFS) parallel file system connection to JUST storage & HPST

JUSUF CPU

- AMD EPYC 7742 (aka 'Rome')
 - x86_64 architecture
 - 64 cores, 2.25 GHz up to 3.4 GHz Boost
 - two-way SMP
- Eight channels of DDR4-3200 memory per socket
 - max bandwidth 190.7 GiB/s per socket
- PCIe 4.0
- 225 Watts TDP

- 2.3 Tflops/s peak performance per node
- **128 CPU cores with 8 memory domains per node**



SYSTEM ACCESS

- Via SSH keys (Ed25519 or RSA 4096-bit) with non-trivial passphrase & 'from' clause
 - installed via JuDOOR account management portal (or automatically by Jupyter-JSC)
- `ssh -X <user>@jusuf.fz-juelich.de`
 - -X option required for remote use of X11-based graphical tools

SOFTWARE ENVIRONMENT

- Rocky Linux 8 OS
- Batch system workload/resource management based on SLURM from ParTec ParaStation
- Programming environment
 - **GNU**, Intel & NVHPC compilers (for C, C++ & Fortran)
 - all supporting OpenMP and other multithreading
 - **ParaStationMPI** (based on MPICH3) & OpenMPI
 - Optimized mathematical libraries (Intel MKL, etc.)
 - CUDA, PAPI, etc.

ACCESSING SOFTWARE

- Hierarchical modules: 'toolchain' constructed by loading compiler then MPI
 - current set Stages/2022
- List available modules (ready to be loaded)
 - `module avail`
- Search for an application/library/tool
 - `module spider <name>`
- Load the desired compiler
 - `module load GCC`
- Load the desired MPI library
 - `module load ParaStationMPI`
- Load additional applications/libraries/tools
 - `module load Extrae Scalasca Vampir`
- List currently loaded modules
 - `module list`
- Purge all loaded modules
 - `module purge`
- Unload an undesired module
 - `module unload CUDA`
- Save current collection of modules
 - `module save [<name>]`
- Restore a saved collection of modules
 - `module restore [<name>]`

FILESYSTEMS

- Jülich Storage cluster (JUST) based on IBM Spectrum Scale (GPFS)
- \$HOME (/p/home/jusers/\$USER)
 - private, regular backup, small storage quota for each user account
- \$PROJECT (/p/project)
 - shared by project members, regular backup, optimized parallel I/O performance
- \$SCRATCH (/p/scratch)
 - shared by project members, no backup, optimized parallel I/O performance, automatic purge based on last file access
- \$DATA, \$FASTDATA, \$LARGEDATA
 - additional storage options for authorized data projects with more demanding needs
- \$ARCHIVE
- ...



ABUSING \$HOME

- \$HOME (/p/home) has neither the capacity nor capability for efficient parallel file I/O
 - building or running any HPC application there is strongly discouraged
 - therefore unsuitable for parallel application trace collection/analysis & storage!
- *\$HOME abuse affects everyone*, and will result in the blocking of your JSC user account



Trademark of Nike, Inc.

BATCH PARTITIONS

- **batch**
 - 1 to 144 standard/slim compute nodes, 24h max
- **gpu**
 - 1 to 46 accelerated compute nodes, 24h max
- **develgpu**
 - 1 to 2 accelerated compute nodes, 24h max
- **training2214** project only has access to batch partition (no GPUs!)
 - reservations for 10 nodes during workshop hours (09:00-16:00)
 - `--reservation=vi-hps-2022-05-17` (Tue)
 - `--reservation=vi-hps-2022-05-18` (Wed)
 - `--reservation=vi-hps-2022-05-19` (Thu)

SLURM

- Show available partitions
 - `sinfo`
- Show queued jobs of your account
 - `squeue -u $USER`
- Cancel queued/running job
 - `scancel <jobid>`
- Submit batch script (to partition)
 - `sbatch <script.sh>`
 - within script ***always*** use `srun` to launch application on allocated compute nodes (rather than the usual `mpiexec/mpiexec` specified by MPI standard)
 - `srun mpi_app.exe`

PROJECT/BUDGET MANAGEMENT

- List your projects
 - `jutil user projects`
- Show your compute allocations/availability
 - `jutil user cpuquota`
- Show your disk quota/usage
 - `jutil user dataquota`
- Activate environment for a particular project
 - `jutil env activate -p <project> [-A <budget>]`

FURTHER INFORMATION / ASSISTANCE

- System information, latest changes, user documentation & FAQs
 - <https://go.fzj.de/JUSUF>
- Cluster status
 - <https://go.fzj.de/status-jusuf-cluster>
- Job execution reports
 - <https://go.fzj.de/llview-jusuf>
- User support helpdesk
 - [mailto: sc@fz-juelich.de](mailto:sc@fz-juelich.de)
 - phone: +49 2461 61-2828

REMOTE SUPERCOMPUTING

1. Unix classic

```
ssh -X jusuf.fz-juelich.de
```

- relies on good network latency/bandwidth and local X11 client

2. Local GUI installations, then

- mount remote filesystem locally:

```
sshfs jusuf.fz-juelich.de:/p/project/training2214 my_jusuf_work
```

or

- explicitly copy required files from remote system:

```
scp [-r] jusuf.fz-juelich.de:/p/project/training2214/<user>/... .
```

3. Use Xpra remote desktop within browser portal Jupyter-JSC

- automatically handles authentication and adjusts connection bandwidth
- (see following slides)



DON'T PANIC!

- Choose whatever approach seems most convenient to get started
- Change later if appropriate

JUPYTER-JSC WEBSERVICE

The image displays three overlapping screenshots of the Jupyter-JSC web service interface. The top screenshot shows a 'Start' page with a 50% progress indicator. The middle screenshot shows a 'Configurations' table. The bottom screenshot shows a landing page with a 'Supercomputing in Your Browser' banner and a 'Jupyter-jsc' section.

| Name | System | Architecture | Project | Partition | Reservation | Resources | Actions |
|------------|--------|--------------|---------|-----------|-------------|-----------|---|
| jupyterlab | JURECA | power8 | compute | compute | | | View Edit |

Jupyter-jsc
Supercomputing in Your Browser

We are pleased to bring "Supercomputing in your browser". Jupyter-jsc is designed to provide the full high performance computing (HPC) ecosystem to the world's most popular software, web browser. JupyterLab is a web based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible to support a wide range of workflows in data science, scientific computing, and machine learning. [Read more](#)

Log in Register

© Forschungszentrum Jülich | [Impressum](#) | [Privacy Policy](#) | [Support](#) | [Terms of Service](#)

HELMHOLTZ
Forschungszentrum Jülich

FIRST TIME LOGIN

0. Pre-requisite: JuDOOR registration with project membership and accepted systems usage agreement
1. Go to link in web browser
<https://jupyter-jsc.fz-juelich.de>
2. Sign in with JSC HPC account (JuDOOR)
3. Register for Jupyter-JSC service
4. Accept usage agreement
5. Submit the registration
6. Wait for email and confirm



CONTROL PANEL

A. Add new JupyterLab config

- Choose a lab name
- Configure the options for the JupyterLab

B. Actions

- Start/delete the named config (workspace will not be deleted)
- Open/stop a running JupyterLab

C. Statusbar

Shows whether systems are online/offline

D. Logout

The screenshot displays the Jülich Supercomputing Centre Control Panel. At the top left is the Jülich logo and 'JÜLICH SUPERCOMPUTING CENTRE'. Below it are 'Start' and 'Links' links. In the top right corner, there is a user profile 'jacob@jsc.juelich.de' and a 'Logout' button, indicated by a red arrow labeled 'D'. The main content area is titled 'Configure A.' and contains instructions: 'Please give each JupyterLab a name. The way you can run multiple JupyterLab on the same time. Supported OS-Images are s-l, u'. Below this is a form to 'name your JupyterLab' and an 'Add new JupyterLab' button. A table lists existing configurations with columns: Name, System, Account/Image, Project, Partition, Reservation, Resou, and Actions. The table contains three rows of data. A red arrow labeled 'B' points to the 'Actions' column. At the bottom, a status bar 'C' shows icons for various systems: jupyter-jsc, JUVELS, JURICA, JUSUF, DEEP, JURON, and HDF-Cloud. The footer includes '© Forschungszentrum Jülich', 'imprint', 'Privacy Policy', 'Support', 'Terms of Service', and the 'HELMHOLTZ' logo.

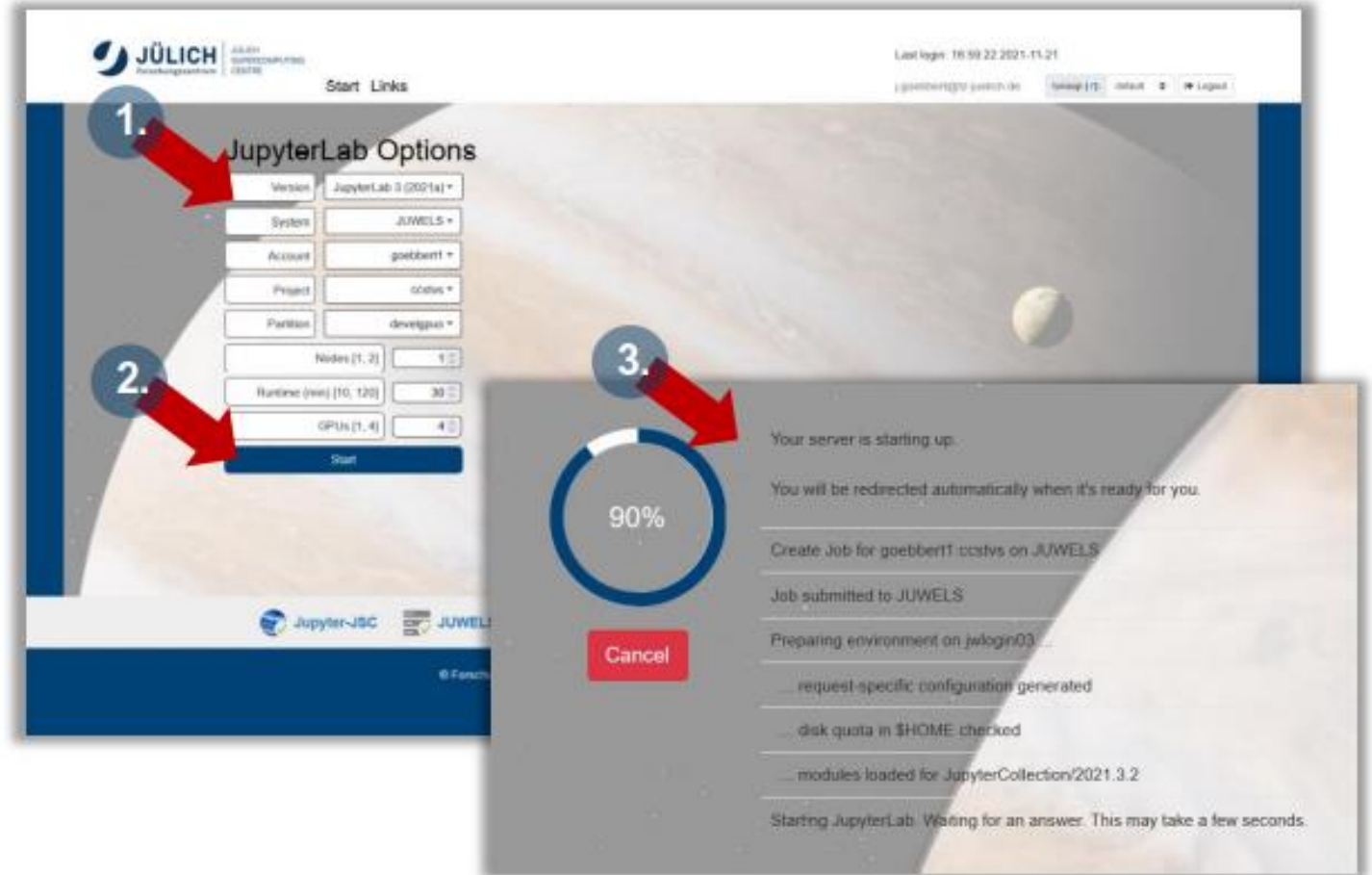
| Name | System | Account/Image | Project | Partition | Reservation | Resou | Actions |
|--------------|--------|---------------|----------------|-----------|-------------|-------|------------|
| jsc123_login | JURICA | gocbeert | compute-system | loginnode | | | start stop |
| jsc456_login | JUSUF | gocbeert | cpu | loginnode | | | start stop |
| jsc789_login | JURICA | gocbeert | cpu | loginnode | | | start stop |

OPTIONS

1. Available options depend on JuDOOR user account settings
 - Version: JupyterLab 3 (default)
 - System: **JUSUF**
 - Account: <your account>
 - Project: **training2214**
 - Partition: LoginNode

(Extra options for Compute partitions)

2. Click on start button (and wait for setup)
 - be patient as it “may take a few seconds”



XPRA

Easy access to a remote desktop

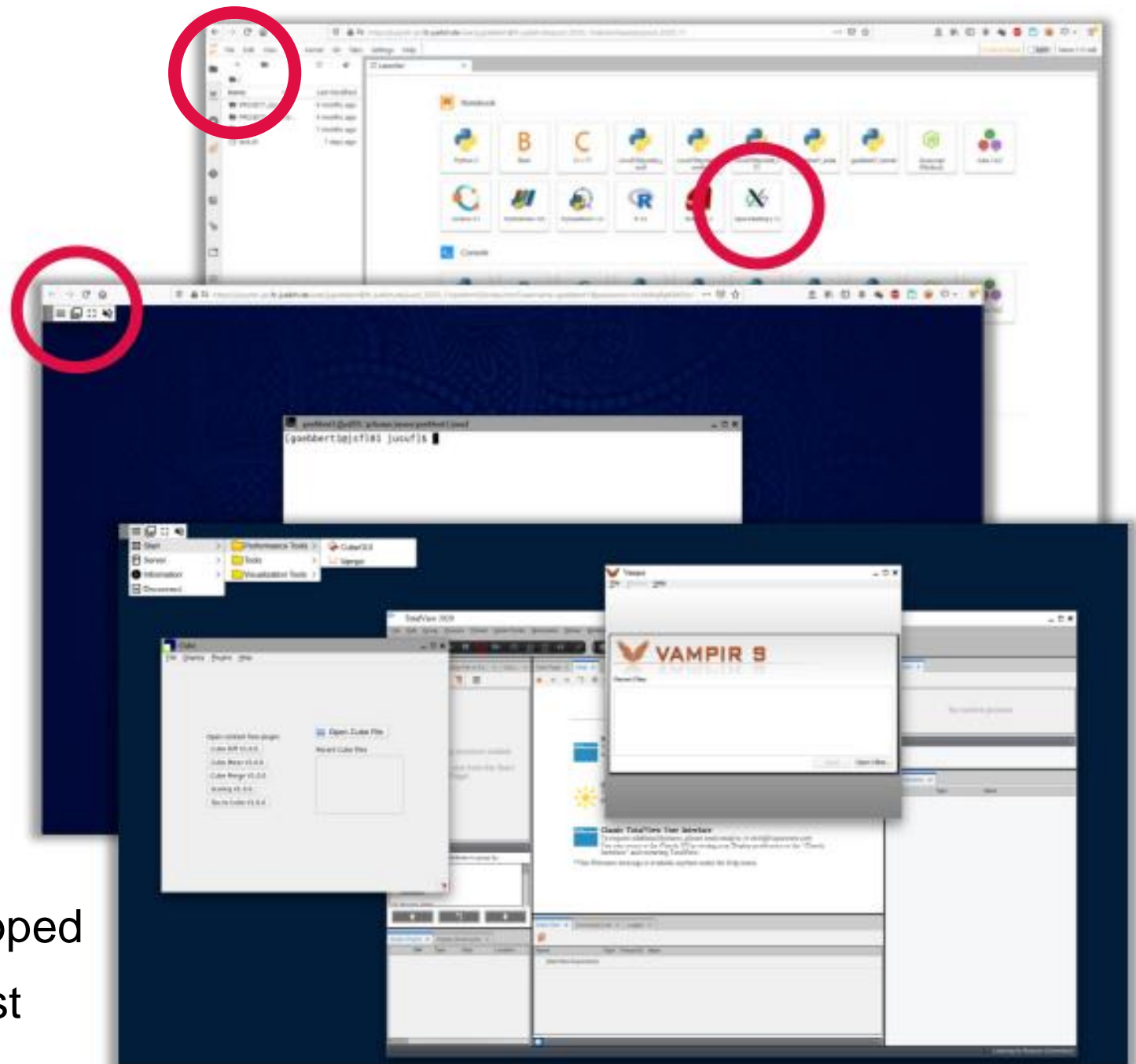
- click '+' launcher then Xpra icon

X Persistent Remote Applications

- runs X clients on a remote host and directs display to local machine
- runs in browser
- reconnection without disrupting the forwarded application session

Remote desktop runs on the same node as JupyterLab does

- killed when JupyterLab session stopped
- refresh browser tab if connection lost



<https://jupyter-jsc.fz-juelich.de>

<https://www.unicore.eu/about-unicore/case-studies/jupyter-at-jsc/>

